

## BMI CALCULATOR

<sup>1</sup>L.S.N.S.NIRMAL KUMAR

<sup>2</sup>Mrs. K R Rajeswari

<sup>1,2</sup>B.V. Raju College Vishnupur::Bhimavaram

### Abstract:

The **Body Mass Index (BMI) Calculator** is a simple Java application designed to determine an individual's BMI based on their weight and height. This program captures user input, processes the data using the BMI formula ( $BMI = weight(kg) / height(m)^2$ ), and categorizes the result into standard health ranges (e.g., underweight, normal weight, overweight, and obese). By implementing clean, modular code and incorporating basic error handling, the calculator ensures accurate results and a user-friendly experience. This tool serves as a practical demonstration of fundamental Java concepts, including input handling, conditional logic, and console output.

## II. PROBLEM STATEMENT

The **existing system** for BMI calculation in Java typically involves a simple console-based or graphical user interface (GUI) application that takes user input for weight and height, processes it using the standard BMI formula ( $BMI = weight(kg) / height(m)^2$ ), and outputs the result along with a classification (underweight, normal, overweight, or obese). Many existing systems rely on basic user input without advanced validation, leading to potential errors if incorrect values are entered. Additionally, most implementations do not store user data for future reference, lack integration with health-tracking systems, and provide only static feedback. While functional, these systems can be enhanced with better validation, historical tracking, and personalized health recommendations.

### 2.1 Existing System Disadvantages

The **existing BMI calculator systems in Java** have several disadvantages, primarily due to their simplicity and lack of advanced features. Most implementations rely on basic user input without proper validation, leading to potential errors if incorrect or unrealistic values are entered. Additionally, they typically do not store user data, making it difficult to track BMI changes over time. Many systems also fail to provide personalized health recommendations based on BMI results, limiting their usefulness beyond simple calculations. Furthermore, most existing implementations lack integration with external health apps or databases, reducing their effectiveness in a comprehensive health monitoring system. Enhancements such as data persistence, better user input validation, and AI-driven health insights could significantly improve these systems.

## III. PROPOSED SYSTEM

The **proposed BMI calculator system in Java** enhances traditional implementations by incorporating improved functionality, accuracy, and user experience. It features robust input validation to prevent errors, ensuring accurate BMI calculations based on weight and height. Unlike existing systems, it includes data storage capabilities to track BMI history over time, allowing users to monitor their health trends. Additionally, the system provides personalized health recommendations based on BMI results, helping users take informed steps toward a healthier lifestyle. Integration with external health applications or databases can further enhance usability, making it a more comprehensive tool. By utilizing a graphical user interface (GUI) or mobile-based implementation, the proposed system improves accessibility and user engagement compared to conventional console-based calculators.

### 3.1 Proposed System Advantages:

The **proposed BMI calculator system in Java** offers several advantages over traditional implementations, enhancing accuracy, usability, and functionality. With robust input validation, it prevents incorrect entries and ensures reliable BMI calculations. The system's ability to store user data allows individuals to track their BMI history over time, providing valuable insights into health trends. Additionally, it offers personalized health recommendations based on BMI results, promoting better decision-making for fitness and wellness. Integration with

external health apps or databases further enhances its usefulness, enabling seamless health monitoring. A user-friendly graphical interface improves accessibility and engagement, making the system more practical for a wide range of users. These enhancements collectively create a more efficient and insightful BMI calculator.

## IV. MODULE DESCRIPTION

There are mainly 6 modules in the project "Housing Society Management " They are:

Admin, User

### 4.1 Admin Account:

Admin login. Admin can check and add various member details. Admin Can view All different category of users.

### 4.2 User:

User can need to enter their name,height and weight then they can get their body mass index value parallely they can able to know the they are belongs to which category(under weight, over weight,normal weight,,,)etc)

## V. REFERENCES

References for the Project Development Were Taken From the following Books and Web Sites.

*JAVA Technologies*

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

JAVA2 Networking by Pistoria

JAVA Security by Scotl oaks

Head First EJB Sierra Bates

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

*HTML*

HTML Black Book by Holzner

*JDBC*

Java Database Programming with JDBC by  
Patel moss.

Software Engineering by Roger Pressman